

## Übungsblatt 3 WT:III

Bis zum 23.05.2018 sind Lösungen zu folgenden Aufgaben abzugeben: 1a,b,c, 2, 3a, 4, 6.

Hinweis: Auf [utilities-online.info/xsdvalidation/](http://utilities-online.info/xsdvalidation/) können Sie Instanzdokumente validieren.

Aufgabe 1 : Modellierung in XML Schema (2+2+1+0 Punkte)

Sie haben den Auftrag Abbildungen wie in wissenschaftlichen Arbeiten (zum Beispiel Abbildung 1) zu modellieren, um diese in XML repräsentieren zu können. Gehen Sie für diese Aufgabe davon aus, dass Abbildungen jeweils genau ein Bild enthalten.

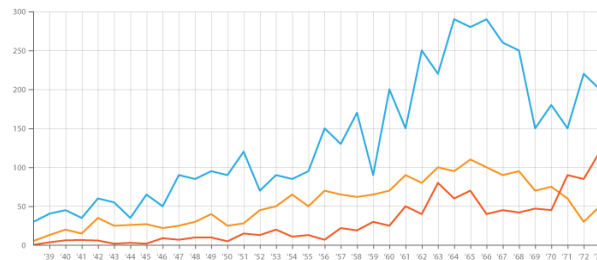


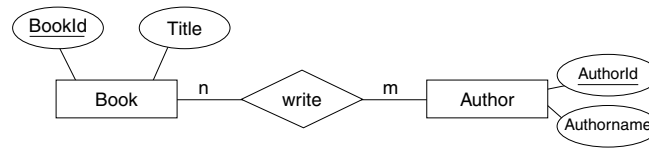
Abbildung 1: Wichtige Ergebnisse.

- (a) Erstellen Sie ein XML-Schema das die folgenden Anforderungen an die Modellierung implementiert:
- Eine Abbildung (<abbildung>) kann (muss aber nicht) eine Abbildungs-Nummer (<nummer>) vom Typ `xsd:integer` haben.
  - Wie in HTML soll das eigentliche Bild (<bild>, der Graph in Fall von Abbildung 1) als leeres Element im Instanzdokument auftreten und die Quelldatei durch ein Pflichtattribut (hier `quelle`) spezifiziert werden.
  - Eine Abbildung enthält genau eine Beschriftung (<beschriftung>, im Beispiel der Text nach „Abbildung 1:“).
- (b) Welche Inhaltsmodelle verwenden jeweils die Elementtypen `abbildung`, `nummer`, `bild` und `beschriftung` in Ihrem Schema?
- (c) Erweitern Sie ihr XML-Schema, so dass es zusätzlich die folgenden Anforderungen an die Modellierung implementiert:
- Ein weiteres Attribut des eigentlichen Bildes soll die Ausrichtung (`ausrichtung`) sein, das die Werte "links", "mittig" oder "rechts" haben kann.
  - Der Defaultwert der Ausrichtung soll "links" sein.
- (d) Erweitern Sie ihr XML-Schema, so dass es zusätzlich die folgenden Anforderungen an die Modellierung implementiert:
- Die Beschriftung kann vor oder nach dem eigentlichen Bild kommen um Bildüberschriften und Bildunterschriften zu unterscheiden.

Geben Sie das Instanzdokument, dass sie zum testen benutzen, mit ab.

Aufgabe 2 : Umsetzung ER-Diagramm zu XML Schema (3+2+2 Punkte)

Setzen Sie das abgebildete ER-Diagramm in XML um.



- (a) Erstellen Sie ein XML-Schema zum ER-Diagramm. Nutzen Sie für die Umsetzung der Fremdschlüsselbeziehungen die Datentypen ID und IDREF.
- (b) Erstellen Sie ein wohlgeformtes XML-Dokument mit mindestens zwei Autoren und einem Buch, das bezüglich des erstellten Schemas valide ist.
- (c) Neben einer Umsetzung mittels ID und IDREF lassen sich Fremdschlüsselbeziehungen in XML Schema auch durch `xsd:key` respektive `xsd:keyref` modellieren. Was sind die wesentlichen Unterschiede in der Modellierung?

Aufgabe 3 : XML Namensräume (1+0 Punkte)

- (a) Wieso ist das folgende XML-Dokument nicht wohlgeformt?

```

<?xml version="1.0"?>
<myNs1:myElem1 xmlns:myNs1="http://www.uni-weimar.de/myNamespace1"
  xmlns:myNs2="http://www.uni-weimar.de/myNamespace1">
  <myNs2:myElem2 myNs1:myAtt="Test1" myNs2:myAtt="Test2">
    myContent
  </myNs2:myElem2>
</myNs1:myElem1>
    
```

- (b) Gegeben sei das folgende XML-Instanzdokument mit Verwendung verschiedener Namensräume:

```

<?xml version="1.0"?>
<pre1:elem1 xmlns:pre1="http://www.example.org/ns1">
  <elem2 att1="w1"/>
  <pre1:elem3 xmlns:pre2="http://www.example.org/ns2">
    <elem4 xmlns="http://www.example.org/ns3">
      <elem5 att2="w2" pre1:att3="w3" pre2:att4="w4"/>
      <pre2:elem6 att5="w5" pre2:att6="w6"/>
      <pre4:elem7 xmlns:pre4="http://www.example.org/ns2"
        pre4:att7="w7" att8="w8"/>
    </elem4>
  </pre1:elem3>
</pre1:elem1>
    
```

Geben Sie die verwendeten Namensräume (Bezeichner oder anonym) an und welche Namen aus diesen Namensräumen verwendet werden. Beispiel:

Namensraum: `http://www.example.org/ns1`  
 Elementnamen: `elem1, elem3`  
 Attributnamen: `att3`

#### Aufgabe 4 : XML Schema: Namensräume in XML-Dokumenten (1+1 Punkte)

Gegeben sei ein XML-Schema für Notizen (note-qua.xsd):

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.uni-weimar.de/note"
  elementFormDefault="qualified">
  <xsd:element name="note">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="to" type="xsd:string"/>
        <xsd:element name="from" type="xsd:string" />
        <xsd:element name="heading" type="xsd:string"/>
        <xsd:element name="body" type="xsd:string"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

Das folgende XML-Dokument beschreibt eine Notiz gemäß des obigen XML-Schemas (note-ns.xml).

```
<?xml version="1.0"?>
<a:note xmlns:a="http://www.uni-weimar.de/note"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.uni-weimar.de/note note-qua.xsd">
  <a:to>Tove</a:to>
  <a:from>Jani</a:from>
  <a:heading>Reminder</a:heading>
  <a:body>Do not forget me this weekend!</a:body>
</a:note>
```

(a) Wie ist das Instanzdokument zu ändern, wenn es mit den nachfolgenden Zeilen beginnt?

```
<?xml version="1.0"?>
<note xmlns="http://www.uni-weimar.de/note"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.uni-weimar.de/note note-qua.xsd">
  ...
```

(b) Wie ist das Instanzdokument zu ändern, wenn das XML-Schema mit den nachfolgenden Zeilen beginnt?

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.uni-weimar.de/note"
  elementFormDefault="unqualified">
  ...
```

#### Aufgabe 5 : Modellierung in XML Schema und Instanziierung

- Entwerfen Sie ein XML-Schema für XML-Dokumente, die Nachrichten beschreiben sollen. Wurzelement sei ein Element 'note'. Das Element soll die Kindelemente 'to', 'from', 'heading' und 'body' besitzen. Der Datentyp der Kindelemente ist 'string'. Außerdem soll das Element 'note' ein Attribut 'delivery' besitzen, das ebenfalls den Datentyp 'string' hat und den Defaultwert 'normal'.
- Weisen Sie dem entworfenen Schema einen Zielnamensraum zu.
- Erstellen Sie ein valides Instanzdokument das ihr Schema referenziert.
- Erläutern Sie die Bedeutung der Schema-Attribute `elementFormDefault` und `attributeFormDefault`.

Aufgabe 6 : Programmiermodul: XML Schema P (4+2+1+1 Punkte)

Schreiben Sie ein XML Schema für einen vereinfachten Atom Feed.

- (a) Schreiben Sie ein Schema für einen einzelnen Feed-Eintrag, zu dem die Datei `entry.xml` von der Kursseite valide ist. Folgende Punkte müssen durch das Schema gegeben sein:
- Jedes Kindelement von `<entry>` darf höchstens einmal vorkommen.
  - Die Kindelemente `<link>` und `<summary>` sind optional, die anderen nicht.
  - Die Reihenfolge ist fest: `<id>`, `<title>`, `<updated>`, `<author>`, `<link>`, `<summary>`.
  - Jedes `<link>`-Element muss genau ein `href`-Attribut haben.
  - Jedes `<author>`-Element muss genau ein `<name>`-Kindelement haben.
- (b) Erweitern Sie das obige Schema für ganze Feeds, so dass die Datei `feed.xml` von der Kursseite ebenfalls valide ist. Die Datei `entry.xml` soll weiterhin valide und die obigen Punkte weiterhin gegeben sein. Folgende Punkte müssen zusätzlich durch das Schema gegeben sein:
- Jedes Kindelement von `<feed>` außer `<entry>` darf höchstens einmal vorkommen.
  - Das Kindelement `<entry>` darf beliebig oft vorkommen.
  - Die Kindelemente `<subtitle>` und `<entry>` sind optional, die anderen nicht.
  - Die Reihenfolge ist fest: `<id>`, `<title>`, `<updated>`, `<author>`, `<subtitle>`, `<entry>`.
  - Die `<author>`-Kindelemente von `<feed>` und `<entry>` sind von demselben Typ.
- (c) Was sind Vor- und Nachteile der Festsetzung der Kindelement-Reihenfolge von `<entry>`?
- (d) Wie kann eine chronologische Reihenfolge der Einträge in der Anzeige garantiert werden, auch wenn XML Schema eine solche Reihenfolge nicht erzwingen kann?

### Hinweise zur Abgabe

- Erstellen Sie *eine* PDF-Datei, in der Texte und Grafiken zu den Aufgaben enthalten sind.
- Quellcode (Java, JavaScript, PHP, HTML, CSS, XML, XSL, XSD, etc.) ist als Textdatei mit entsprechender Dateiendung zu übersenden; Quellcode innerhalb des PDF-Dokuments wird nicht korrigiert.
- Referenzieren Sie Ihre Quellcode-Dateien in dem PDF-Dokument, so dass sie einer Aufgabe eindeutig zuzuordnen sind.
- Verpacken Sie Ihre Abgabe in *einer* ZIP-Datei die wie folgt benannt ist:  
`<Nachname>-<MatrikelNr>-webtec-blatt<Übungsblattnummer>.zip`.
- Gruppenabgaben mit bis zu drei Personen sind erlaubt; pro Gruppe genügt dann eine Abgabe, in der alle Gruppenmitglieder mit Namen, Matrikel, und E-Mail-Adresse verzeichnet sind
- Senden Sie die ZIP-Datei an Johannes Kiesel ([johannes.kiesel@uni-weimar.de](mailto:johannes.kiesel@uni-weimar.de)).